[4] K. Steiglitz and L. F. McBride, "A technique for the identification of linear systems," *IEEE Trans. Automat. Contr.,* vol. AC-10, pp. 461–464, July 1965.

[5] J. E. Cousseau and P. S. R. Diniz, "A consistent Steiglitz–McBride algorithm," in *IEEE Proc. Int. Symp. Circ. Syst.,* Chicago, IL, 1993, pp. 52–55.

[6] L. Pandya, "A class of bootstrap estimators and their relationship to the generalized two stage least squares estimators," *IEEE Trans. Automat. Contr.,* vol. AC-19, pp. 831–835, Dec. 1974.

[7] P. Stoica and T. Söderström, "Asymptotic behavior of some bootstrap estimators," *Int. J. Contr.,* vol. 33, pp. 433–454, June 1981.

[8] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

A Minimal, Rotation-Based FRLS Lattice Algorithm

François Desbouvries and Phillip Regalia

Abstract—We propose an alternate Givens rotation-based least-squares lattice algorithm. Based on spherical trigonometry principles, this algorithm turns out to be a normalized version of the fast QRD-based least-squares lattice filter, introduced independently by Ling and by Proudler *et al*. In contrast with that algorithm, the storage requirements of the new algorithm are minimal (in the system theory sense). From this, we show that the new algorithm satisfies the backward consistency property and, hence, enjoys stable error propagation.

I. INTRODUCTION

Yule's PARCOR identity was recently shown to coincide with the cosine law of spherical trigonometry. This observation establishes a connection between fast recursive least-squares (FRLS) adaptive filtering and spherical trigonometry [1], [2] because the prewindowed fully normalized FRLS lattice algorithm of Lee *et al.* [3] consists of three particular applications of Yule's PARCOR identity. Let us briefly recall this algorithm for convenience of the reader. Let $\nu_{n,t}$ and $\eta_{n,t}$ be the doubly normalized forward and backward prediction errors, respectively, at order n and time t, and let $\rho_{n+1,t}$ be the $n + 1$st-order PARCOR at time t. (We restrict ourselves to scalar signals for simplicity). The algorithm of Lee *et al.* consists of the well-known recursions

$$\rho_{n+1,t} = \nu_{n,t}\eta_{n,t-1} + \sqrt{1 - \nu_{n,t}^2}\,\rho_{n+1,t-1}\sqrt{1 - \eta_{n,t-1}^2}$$

$$\nu_{n+1,t} = (1 - \rho_{n+1,t}^2)^{-1/2}(\nu_{n,t} - \rho_{n+1,t}\eta_{n,t-1})$$
$$\cdot (1 - \eta_{n,t-1}^2)^{-1/2}$$

$$\eta_{n+1,t} = (1 - \rho_{n+1,t}^2)^{-1/2}(\eta_{n,t-1} - \rho_{n+1,t}\nu_{n,t})$$
$$\cdot (1 - \nu_{n,t}^2)^{-1/2}. \tag{1}$$

Recast in the spherical trigonometry framework [1], [2], the six PARCOR's propagated by this algorithm are the cosines of the six elements of a spherical triangle, namely, the angles $A, B,$ and C and their corresponding sides $a, b,$ and c; the correspondences are $\cos A = \rho_{n+1,t-1}, \cos b = \nu_{n,t}, \cos c = \eta_{n,t-1}, \cos a =$

$\rho_{n+1,t}, \cos B = \nu_{n+1,t},$ and $\cos C = \eta_{n+1,t}.$ Thus, (1) can be seen as one particular solution to the spherical triangle problem that, given two sides b and c of a spherical triangle plus the angle A in between, consists of determining the remaining three elements $a, B,$ and $C.$

It turns out that this algorithm (using only the cosine law three times) is not, in spherical trigonometry, a standard solution of the spherical triangle problem $(A, b, c) \to (a, B, C).$ Among other alternatives, one classical way for (partially) solving the problem consists in using a so-called "Gauss system" [4]–[6]. In this correspondence, we obtain a new algorithm based on two such Gauss systems, which yields, in effect, a normalized version of the QRD-based LS lattice filter, introduced independently by Ling [7] and Proudler *et al.* [8], [9]. The main advantage of our algorithm over that of [7]–[9] is that the storage requirements are cut in half, thereby purging the "redundancy" present in that algorithm. Indeed, the new algorithm now propagates a minimal (in the system theory sense) number of internal variables, all of magnitude bounded by 1. Now, minimality is known to be a key structural constraint behind backward consistency [10], and from this viewpoint, we prove that this algorithm enjoys stable error propagation.

This correspondence is organized as follows. In Section II, we give a geometric interpretation of the fast QRD-based lattice filter in terms of spherical trigonometry. In this framework, it becomes obvious that some appropriate normalization should be performed, which leads to our new algorithm in Section III. Stable error propagation is proved in Section IV, with concluding remarks in Section V.

II. SPHERICAL COORDINATE TRANSFORMATIONS, GAUSS SYSTEMS, AND LEAST-SQUARES LATTICE FILTERS

Consider a unit sphere in 3-D space centered at the point $O.$ Let C' be an arbitrary point, and consider a line beginning at O and passing through $C',$ with C its intersection with the unit sphere. The position of C' can be described, for example, by the spherical coordinates $(R, \theta_1, \varphi_1),$ where R is the length of the vector $OC',$ and the two angles satisfy $\theta_1 \in [-\pi, \pi]$ and $\varphi_1 \in [-\pi/2, \pi/2]$ or, alternately, by the Cartesian coordinates $(x_1 = R\cos\varphi_1\cos\theta_1, y_1 = R\cos\varphi_1\sin\theta_1, z_1 = R\sin\varphi_1)$ (see Fig. 1).

Let us rotate the initial coordinate system (O, x_1, y_1, z_1) by an angle c around the y_1 axis, giving a new coordinate system $(O, x_2, y_2, z_2),$ with $y_2 = y_1.$ The coordinates of C' in the new system are $(x_2 = R\cos\varphi_2\cos\theta_2, y_2 = R\cos\varphi_2\sin\theta_2, z_2 = R\sin\varphi_2).$ Let us now write the mapping $[x_1, y_1, z_1]^T \xrightarrow{c} [x_2, y_2, z_2]^T$ in terms of the elements of the spherical triangle $ABC,$ using the relations $a = \pi/2 - \varphi_2, b = \pi/2 - \varphi_1, A = \pi - |\theta_1|,$ and $B = |\theta_2|.$ If we let ϵ denote $\theta_1/|\theta_1| = \theta_2/|\theta_2|,$ we get a "Gauss system" (unnormalized if $R \neq 1$)

$$\left(\begin{bmatrix} \cos B \sin a \\ \epsilon \sin B \sin a \\ \cos a \end{bmatrix} R \right) = \begin{bmatrix} \cos c & 0 & \sin c \\ 0 & 1 & 0 \\ -\sin c & 0 & \cos c \end{bmatrix} \left(\begin{bmatrix} -\cos A \sin b \\ \epsilon \sin A \sin b \\ \cos b \end{bmatrix} R \right) \tag{2}$$

whose three rows yield, respectively, the five-element formula, the sine law, and the cosine law of spherical trigonometry. These equations appear naturally in astronomy [4]–[6]. Indeed, passing back and forth between the horizontal coordinates of a star and its hour coordinates, or between its equatorial coordinates and its ecliptic coordinates, consists of performing such a change of spherical coordinates $(\theta_1, \varphi_1) \xrightarrow{c} (\theta_2, \varphi_2)$; this can be done using (2) since,
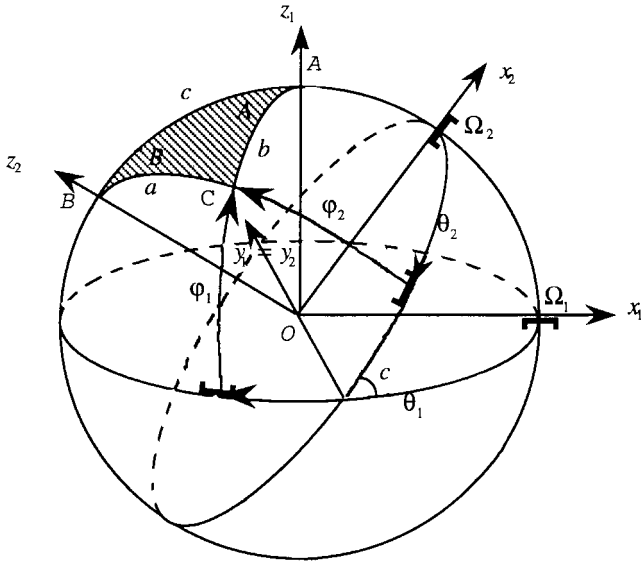
Fig. 1. Illustrating the spherical coordinate system.

in these applications, one is interested only in solving the partial problem $(A, b, c) \rightarrow (a, B)$.

These equations also arise in adaptive filtering and may be linked via the classical geometric framework [11] of FRLS algorithms [2]. Let $A, B,$ and C belong to the space $\mathbb{R}^{t+1}$ of $(t + 1)$-dimensional vectors, with inner product $\langle u, v \rangle = u^T \Lambda v$, where $\Lambda = \text{diag}\{\lambda^t, \cdots, \lambda, 1\}$ with λ the forgetting factor. Let Y be a data matrix with $t + 1$ rows, and let $P_Y^\perp = I - Y(Y^T \Lambda Y)^{-1} Y^T \Lambda$ be the projection matrix onto the orthogonal complement subspace to the column space of Y. Let us identify $\vec{OA}$ with $\overline{P_Y^\perp A}, \vec{OB}$ with $\overline{P_Y^\perp B}, \vec{OC}$ with $\overline{P_Y^\perp C}$, and $\vec{OC'}$ with $P_Y^\perp C$, where $\overline{P_Y^\perp C} = P_Y^\perp C \langle P_Y^\perp C, P_Y^\perp C \rangle^{-1/2}$ designates scaling to unit length. Last, let the PARCOR $\rho_Y(u, v)$ be defined as $\langle \overline{P_Y^\perp u}, \overline{P_Y^\perp v} \rangle$. The PARCOR analogs to the first and third rows of (2) are [2]

$$\begin{bmatrix} \rho_{Y,B}(A, C) \sqrt{1 - \rho_Y^2(B, C)} \\ \rho_Y(B, C) \end{bmatrix} \langle P_Y^\perp C, P_Y^\perp C \rangle^{1/2}$$
$$= \begin{bmatrix} \rho_Y(A, B) & \sqrt{1 - \rho_Y^2(A, B)} \\ -\sqrt{1 - \rho_Y^2(A, B)} & \rho_Y(A, B) \end{bmatrix}$$
$$\cdot \begin{bmatrix} -\rho_{Y,A}(B, C) \sqrt{1 - \rho_Y^2(A, C)} \\ \rho_Y(A, C) \end{bmatrix} \langle P_Y^\perp C, P_Y^\perp C \rangle^{1/2}. \quad (3)$$

Now, let $\{y_i\}$ and $\{d_i\}$ denote the input and desired-response time series, respectively, of the adaptive filter. Let $\mathbf{y}_i \overset{\text{def}}{=} [0 \cdots 0 \ y_0 \cdots y_i]^T$, $\mathbf{d}_i \overset{\text{def}}{=} [0 \cdots 0 \ d_0 \cdots d_i]^T$, and $\sigma \overset{\text{def}}{=} [0 \cdots 0 \ 1]^T$ belong to $\mathbb{R}^{t+1}$, and set Y to the $(t + 1) \times n$ matrix $[\mathbf{y}_{t-1} \cdots \mathbf{y}_{t-n}]$. The successive correspondences $(A = \sigma, B = \mathbf{y}_t, C = \mathbf{y}_{t-n-1}), (A = \sigma, B = \mathbf{y}_{t-n-1}, C = \mathbf{y}_t)$ and $(A = \sigma, B = \mathbf{y}_{t-n-1}, C = \mathbf{d}_{t-1})$ yield, respectively

$$\begin{bmatrix} \nu_{n,t} & \sqrt{1 - \nu_{n,t}^2} \\ -\sqrt{1 - \nu_{n,t}^2} & \nu_{n,t} \end{bmatrix} \begin{bmatrix} -\sqrt{\lambda} \rho_{n+1,t-1} \sqrt{E_n^b(t-2)} \\ e_{n,t-1}^b / \sqrt{\gamma_n(t-1)} \end{bmatrix}$$
$$= \begin{bmatrix} e_{n+1,t}^b / \sqrt{\gamma_{n+1}(t)} \\ \rho_{n+1,t} \sqrt{E_n^b(t-1)} \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \eta_{n,t-1} & \sqrt{1 - \eta_{n,t-1}^2} \\ -\sqrt{1 - \eta_{n,t-1}^2} & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\sqrt{\lambda} \rho_{n+1,t-1} \sqrt{E_n^f(t-1)} \\ e_{n,t}^f / \sqrt{\gamma_n(t-1)} \end{bmatrix}$$
$$= \begin{bmatrix} e_{n+1,t}^f / \sqrt{\gamma_{n+1}(t-1)} \\ \rho_{n+1,t} \sqrt{E_n^f(t)} \end{bmatrix} \quad (5)$$

$$\cdot \begin{bmatrix} \eta_{n,t-1} & \sqrt{1 - \eta_{n,t-1}^2} \\ -\sqrt{1 - \eta_{n,t-1}^2} & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\sqrt{\lambda} \pi_{n,t-2} \\ e_{n,t-1} / \sqrt{\gamma_n(t-1)} \end{bmatrix}$$
$$= \begin{bmatrix} e_{n+1,t-1} / \sqrt{\gamma_{n+1}(t-1)} \\ \pi_{n,t-1} \end{bmatrix}. \quad (6)$$

Here, $e_{n,t}^f, e_{n,t}^b$ and $e_{n,t}$ are, respectively, the unnormalized *a posteriori* forward prediction error, backward prediction error, and filtering error, at order n and time t. The term $E_n^f(t) \overset{\text{def}}{=} \langle P_Y^\perp \mathbf{y}_t, P_Y^\perp \mathbf{y}_t \rangle$ (resp. $E_n^b(t - 1) \overset{\text{def}}{=} \langle P_Y^\perp \mathbf{y}_{t-n-1}, P_Y^\perp \mathbf{y}_{t-n-1} \rangle$) is the forward (resp. backward) prediction error energy. The term $\gamma_n(t - 1) \overset{\text{def}}{=} \langle P_Y^\perp \sigma, P_Y^\perp \sigma \rangle$ is a "likelihood variable," and $\pi_{n,t-1} = \langle P_Y^\perp \mathbf{d}_{t-1}, \overline{P_Y^\perp \mathbf{y}_{t-n-1}} \rangle$ is an auxiliary variable. We recognize in (4) and (5) the rotations of the prediction part, and in (6) that of the filtering part, of the fast QRD-based lattice [7]—[9], which is thus essentially made of three unnormalized Gauss systems.

III. A MINIMAL GIVENS ROTATION-BASED LATTICE ALGORITHM

The algorithm above is nonminimal in its storage requirements, which impedes a direct verification of stable error propagation (e.g., [12]). In effect, the parameters $\nu_{n,t}$ and $\sqrt{1 - \nu_{n,t}^2}$ of the forward rotation are determined according to

$$\sqrt{E_n^f(t)} = [(\sqrt{\lambda E_n^f(t-1)})^2 + (e_{n,t}^f / \sqrt{\gamma_n(t-1)})^2]^{1/2} \quad (7)$$

$$\nu_{n,t} = \frac{e_{n,t}^f / \sqrt{\gamma_n(t-1)}}{\sqrt{E_n^f(t)}}, \quad \sqrt{1 - \nu_{n,t}^2} = \frac{\sqrt{\lambda E_n^f(t-1)}}{\sqrt{E_n^f(t)}}. \quad (8)$$

Similarly, $\eta_{n,t-1}$ and $\sqrt{1 - \eta_{n,t-1}^2}$ are determined according to

$$\sqrt{E_n^b(t-1)} = [(\sqrt{\lambda E_n^b(t-2)})^2 + (e_{n,t-1}^b / \sqrt{\gamma_n(t-1)})^2]^{1/2} \quad (9)$$

$$\eta_{n,t-1} = \frac{e_{n,t-1}^b / \sqrt{\gamma_n^{(t-1)}}}{\sqrt{E_n^b(t-1)}},$$

$$\sqrt{1 - \eta_{n,t-1}^2} = \frac{\sqrt{\lambda E_n^b(t-2)}}{\sqrt{E_n^b(t-1)}}. \quad (10)$$

Thus, this algorithm requires storing $5M + (M + 1)$ variables at each iteration (where M is the maximum order of the filter), namely, the variables $\left\{\sqrt{E_n^f(t-1)}\right\}_{n=0}^{M-1}, \left\{\sqrt{E_n^b(t-2)}\right\}_{n=0}^{M-1}, \left\{e_{n,t-1}^b / \sqrt{\gamma_n(t-1)}\right\}_{n=0}^{M-1}, \{\rho_{n+1,t-1} \sqrt{E_n^f(t-1)}\}_{n=0}^{M-1}$, and $\{\rho_{n+1,t-1} \sqrt{E_n^b(t-2)}\}_{n=0}^{M-1}$ for the prediction section, as well as $\{\pi_{n,t-2}\}_{n=0}^{M}$ for the filtering part.

Now, in many applications, the quantity of ultimate interest is often the filtered error $e_{M+1,t-1}$, which is computed from $e_{M+1,t-1} / \sqrt{\gamma_{M+1}(t-1)}$ and $\sqrt{\gamma_{M+1}(t-1)}$, where $\sqrt{\gamma_{M+1}(t-1)}$ is given by

$$\sqrt{\gamma_{n+1}(t-1)} = \sqrt{1 - \eta_{n,t-1}^2} \sqrt{\gamma_n(t-1)}. \quad (11)$$

We now show that in such cases, the storage requirements of the fast QRD-based least-squares lattice can be cut in half. In particular, the vectors to which the rotations (4) and (5) are applied are closely related to the parameters of these rotations so that it is not in fact necessary to compute [as in (8) and (10)] the parameters of these rotations. More precisely, (4) and (5) can be replaced by the corresponding *normalized* Gauss systems, i.e., the algorithm that

TABLE I
MINIMAL LATTICE ALGORITHM

Initialization:

$$\eta_{n,-1} = \pi_{n,-1} = \rho_{n+1,-1} = 0, \quad \text{for } n = 0, 1, \ldots, M-1;$$

$$\sqrt{E_0^f(-1)} = 0 \text{ (exact start) or } \sqrt{E_0^f(-1)} = \delta > 0 \text{ (soft start)}$$

for time $t = 0, 1, 2, \ldots$:

$$\gamma_0(t) = 1, \quad e_{0,t} = d_t, \quad \sqrt{E_0^f(t)} = \left\| \begin{matrix} \lambda^{1/2}\sqrt{E_0^f(t-1)} \\ y_t \end{matrix} \right\|, \quad v_{0,t} = \eta_{0,t} = \frac{y_t}{\sqrt{E_0^f(t)}}$$

$$\sqrt{\gamma_1(t)} = \sqrt{1 - \eta_{0,t}^2}$$

$$\begin{bmatrix} \eta_{0,t} & \sqrt{1-\eta_{0,t}^2} \\ -\sqrt{1-\eta_{0,t}^2} & \eta_{0,t} \end{bmatrix} \begin{bmatrix} -\lambda^{1/2}\pi_{0,t-1} \\ e_{0,t} \end{bmatrix} = \begin{bmatrix} e_{1,t}/\sqrt{\gamma_1(t)} \\ \pi_{0,t} \end{bmatrix}$$

for order $n = 0, \ldots, M-1$:

prediction:

$$\begin{bmatrix} \eta_{n,t-1} & \sqrt{1-\eta_{n,t-1}^2} \\ -\sqrt{1-\eta_{n,t-1}^2} & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1}\sqrt{1-v_{n,t}^2} \\ v_{n,t} \end{bmatrix} = \begin{bmatrix} v_{n+1,t}\sqrt{1-\rho_{n+1,t}^2} \\ \rho_{n+1,t} \end{bmatrix}$$

$$\begin{bmatrix} v_{n,t} & \sqrt{1-v_{n,t}^2} \\ -\sqrt{1-v_{n,t}^2} & v_{n,t} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1}\sqrt{1-\eta_{n,t-1}^2} \\ \eta_{n,t-1} \end{bmatrix} = \begin{bmatrix} \eta_{n+1,t}\sqrt{1-\rho_{n+1,t}^2} \\ * \end{bmatrix}$$

$$v_{n+1,t} = \left(v_{n+1,t}\sqrt{1-\rho_{n+1,t}^2} \right) \Big/ \sqrt{1-\rho_{n+1,t}^2}$$

$$\eta_{n+1,t} = \left(\eta_{n+1,t}\sqrt{1-\rho_{n+1,t}^2} \right) \Big/ \sqrt{1-\rho_{n+1,t}^2}$$

filtering:

$$\sqrt{\gamma_{n+2}(t)} = \sqrt{1-\eta_{n+1,t}^2}\sqrt{\gamma_{n+1}(t)}$$

$$\begin{bmatrix} \eta_{n+1,t} & \sqrt{1-\eta_{n+1,t}^2} \\ -\sqrt{1-\eta_{n+1,t}^2} & \eta_{n+1,t} \end{bmatrix} \begin{bmatrix} -\lambda^{1/2}\pi_{n+1,t-1} \\ e_{n+1,t}/\sqrt{\gamma_{n+1}(t)} \end{bmatrix} = \begin{bmatrix} e_{n+2,t}/\sqrt{\gamma_{n+2}(t)} \\ \pi_{n+1,t} \end{bmatrix}$$

end for order loop

$$e_{M+1,t} = \left(e_{M+1,t}/\sqrt{\gamma_{M+1}(t)} \right) \sqrt{\gamma_{M+1}(t)}$$

end for time loop.

---

transforms the coordinates of $\overline{P_Y^\perp C}$, rather than $P_Y^\perp C$:

$$\begin{bmatrix} \nu_{n,t} & \sqrt{1-\nu_{n,t}^2} \\ -\sqrt{1-\nu_{n,t}^2} & \nu_{n,t} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1}\sqrt{1-\eta_{n,t-1}^2} \\ \eta_{n,t-1} \end{bmatrix}$$

$$= \begin{bmatrix} \eta_{n+1,t}\sqrt{1-\rho_{n+1,t}^2} \\ \rho_{n+1,t} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \eta_{n,t-1} & \sqrt{1-\eta_{n,t-1}^2} \\ -\sqrt{1-\eta_{n,t-1}^2} & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1}\sqrt{1-\nu_{n,t}^2} \\ \nu_{n,t} \end{bmatrix}$$

$$= \begin{bmatrix} \nu_{n+1,t}\sqrt{1-\rho_{n+1,t}^2} \\ \rho_{n+1,t} \end{bmatrix}. \tag{13}$$

The algorithm made of (12) and (13) for the prediction part, and keeping (6) and (11) for the filtering part, is summarized in Table I; the storage requirements of the prediction part are reduced to $2M+1$ parameters (the minimal value [10]), namely, the square-root signal energy $\sqrt{E_0^f(t)}$, the doubly normalized backward prediction error

residuals $\{\eta_{n,t-1}\}_{n=0}^{M-1}$, and the PARCOR's $\{\rho_{n+1,t-1}\}_{n=0}^{M-1}$. The filtering part still uses the $M+1$ parameters $\{\pi_{n,t-1}\}_{n=0}^{M}$.

This algorithm is a rotation version of that suggested in [2, Section E]. As each prediction stage of our algorithm performs the mapping $(\rho_{n+1,t-1}, \nu_{n,t}, \eta_{n,t-1}) \rightarrow (\rho_{n+1,t}, \nu_{n+1,t}, \eta_{n+1,t})$, it is clearly related to (1), as well as to the algorithm of [13, (10) and (11)], which perform the same global transformation. This can be explained by spherical trigonometry principles: Since there are three degrees of freedom in a spherical triangle, there can be no more that three distinct relationships among the six elements. One such set is comprised of the three cosine laws, i.e., (1) in the FRLS framework, from which any other spherical trigonometry formula may be deduced.

IV. STABLE ERROR PROPAGATION

The verification of stable error propagation follows the methodology of [10], in which stable error propagation is shown subordinate to a backward consistency property. The time updates in Table I may

be written as a state recursion

$$\xi(t) = T[\xi(t-1), y_t] \quad (14)$$

in which

$$\xi(t) = [\sqrt{E_0^f(t)}; \eta_{0,t}; \cdots; \eta_{M-1,t}; \rho_{1,t}; \cdots; \rho_{M,t}] \quad (15)$$

is the state vector comprising the variables that need be written for storage at each iteration, and $T[\cdot, \cdot]$ is the mapping that implements the subroutine of Table I at each iteration.

As in [10] and [12], we denote by $\mathcal{S}_i$ the set of reachable states $\xi(t)$ in exact arithmetic, as the past input $y_t, y_{t-1}, y_{t-2}, \cdots$ is varied over all possibilities. If the computed state vector, call it $\tilde{\xi}(t)$ (including thus roundoff errors), satisfies

$$\tilde{\xi}(t) \in \mathcal{S}_i \quad (16)$$

then $\tilde{\xi}(t)$ may be understood as the exact solution reached by some perturbed past input sequence $\tilde{y}_t, \tilde{y}_{t-1}, \tilde{y}_{t-2}, \cdots$. The influence of the perturbation $\tilde{\xi}(t) - \xi(t)$ on the future evolution of (14) is then identical to the influence of a corresponding perturbation on the past input sequence. With persistently exciting future data $y_{t+1}, y_{t+2}, \cdots$, the influence of this perturbation will be exponentially decaying owing to exponential data weighting ($\lambda < 1$). For more detail on this simple argument, we refer the reader to [10], [12], [14], and [15]. It thus suffices to examine whether the consistency condition (16) can be enforced in the present algorithm.

To this end, we can connect the variables of the state vector (15) to the variables of the QR algorithm studied in [10]. The PARCORS $\{\rho_{n,t}\}$ correspond, in the notations of [10], to $\rho_{n,t} = \sin \phi_n(t)$, whereas the doubly normalized backward prediction errors become $\eta_{n,t} = \sin \theta_n(t)$. From [10, Prop. 5], the set of reachable state variables is characterized by the simple inequalities $|\theta_{n-1}(t)| < \pi/2$ and $|\phi_n(t)| < \pi/2$ for orders $n = 0, 1, \cdots, M$, combined with one prediction error energy being positive. This yields $\mathcal{S}_i$ in the present algorithm as

$$|\eta_{n-1,t}| < 1, \quad |\rho_{n,t}| < 1, \quad \text{and} \quad \sqrt{E_0^f(t)} > 0 \quad (17)$$

for all n and t. Provided the computed variables always satisfy these inequalities, consistency applies, which then implies stable error propagation [10], [12], [14], [15].

To examine satisfaction of the consistency constraints (17), assume that the expression holds at time t, i.e., $|\eta_{n-1,t}| < 1$ and $|\rho_{n,t}| < 1$; we then examine whether it remains in effect at time $t+1$. The first computation reads as

$$\sqrt{E_0^f(t+1)} = \left\| \begin{matrix} \lambda^{1/2} \sqrt{E_0^f(t)} \\ y_{t+1} \end{matrix} \right\| > 0,$$

$$\eta_{0,t+1} = \nu_{0,t+1} = \frac{y_{t+1}}{\sqrt{E_0^f(t+1)}}$$

$$\Rightarrow |\eta_{0,t+1}| = |\nu_{0,t+1}| < 1.$$

(The intermediate variables $\nu_{n,t}$ intervene in the computation $\xi(t+1) = T[\xi(t), y_{t+1}]$ but are not written for storage). The computation (12) becomes

$$\begin{bmatrix} \nu_{n-1,t+1} & \sqrt{1 - \nu_{n-1,t+1}^2} \\ -\sqrt{1 - \nu_{n-1,t+1}^2} & \nu_{n-1,t+1} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} -\rho_{n,t}\sqrt{1 - \eta_{n-1,t}^2} \\ \eta_{n-1,t} \end{bmatrix}}_{\stackrel{\text{def}}{=} a} = \underbrace{\begin{bmatrix} \eta_{n,t+1}\sqrt{1 - \rho_{n,t+1}^2} \\ \rho_{n,t+1} \end{bmatrix}}_{\stackrel{\text{def}}{=} b}.$$

Now, with $|\rho_{n,t}| < 1$ and $|\eta_{n-1,t}| < 1$, we get $\|a\| < 1$. If the norm of the numerical rotation on the left-hand side is less than $1/\|a\|$, we

obtain $\|b\| < 1$. This holds, in particular, if the rotation is numerically passive. The property $\|b\| < 1$ then ensures that the inferred variables $\eta_{n,t+1}$ and $\rho_{n,t+1}$ fulfill $|\eta_{n,t+1}| < 1$ and $|\rho_{n,t+1}| < 1$ so that these computed variables remain consistent. The same argument applied to (13) shows that $|\nu_{n,t+1}| < 1$ as well so that the rotation required by the next order index is well defined. Thus, if $\xi(t) \in \mathcal{S}_i$, then the updated state vector so computed likewise satisfies $\xi(t+1) \in \mathcal{S}_i$. This gives backward consistency and, thus, stable error propagation.

V. CONCLUDING REMARKS

An alternate rotation-based lattice algorithm was derived. In contrast with the algorithm of Ling [7] and Proudler *et al.* [8], [9], the present algorithm is minimal in the storage requirements. From this, stable error propagation is shown easily on verifying that the computed variables remain backward consistent. The consistency test of the algorithm amounts to verifying that the square-root signal energy remains positive and that all other variables are bounded by unit magnitude. This can be structurally induced by correct programming or at least tested with negligible overhead if less careful programming methods are employed.

Our contribution extends the family of minimal fast least-squares algorithms beyond those identified in [10] and [14]; in contrast to the backward consistent algorithm of [10], for which the order recursions are both ascending and descending, all recursions in the present algorithm are ascending, which thereby affords pipelining of the computations.

REFERENCES

[1] F. Desbouvries, "Recursive least-squares lattices and trigonometry in the spherical triangle," in *Proc. ICASSP*, Minneapolis, MN, Apr. 27–30, 1993, pp. III.404–III.407.

[2] ——, "Spherical trigonometry, Yule's PARCOR identity and FRLS fully-normalized lattice," *IEEE Trans. Signal Processing*, vol. 44, pp. 427–430, Feb. 1996.

[3] D. T. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, pp. 627–641, June 1981.

[4] V. Kourganoff, *Astronomie Fondamentale Élémentaire*. Paris: Masson, 1961.

[5] H. Andoyer, *Cours d'Astronomie*, 3rd ed. Paris: Librairie Scientifique J. Hermann, 1923.

[6] W. Chauvenet, *A Treatise on Plane and Spherical Trigonometry*, 9th ed. Philadelphia, PA: Lippincott, 1875.

[7] F. Ling, "Givens rotation based least squares lattice and related algorithms," *IEEE Trans. Signal Processing*, vol. 39, pp. 1541–1551, 1991.

[8] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "Fast QRD-based algorithms for least squares linear prediction," in *Mathematics in Signal Processing II*, J. G. McWhirter, Ed. Oxford, U.K.: Clarendon, 1990, pp. 465–488.

[9] ——, "Computationally efficient QR decomposition approach to least squares adaptive filtering," *Proc. Inst. Elec. Eng.*, pt. F, vol. 138, pp. 341–353, Aug. 1991.

[10] P. A. Regalia, "Numerical stability properties of a QR-based fast least-squares algorithm," *IEEE Trans. Signal Processing*, vol. 41, pp. 2096–2109, June 1993.

[11] H. Lev-Ari, T. Kailath, and J. Cioffi, "Least-squares adaptive lattice and transversal filters: A unified geometric theory," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 222–236, Mar. 1984.

[12] P. A. Regalia, "Numerical stability issues in fast least-squares adaptation algorithms," *Opt. Eng.*, vol. 31, pp. 1144–1152, June 1992.

[13] H. M. Ahmed, J. M. Delosme, and M. Morf, "Highly concurrent computing structures for matrix arithmetic and signal processing," *Comput.*, vol. 15, pp. 65–80, Jan. 1982.

[14] D. T. M. Slock, "The backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms," *Opt. Eng.*, vol. 31, pp. 1153–1169, June 1992.

[15] P. A. Regalia and F. Desbouvries, "Displacement structures of covariance matrices, lossless systems, and numerical algorithm design," *SIAM J. Matrix Anal. Applications*, vol. 16, pp. 536–564, Apr. 1995.